

Winter 1-15-2019

# Development of a Personal Logbook Android Application

John Byrnes

La Salle University, byrnesj1@student.lasalle.edu

Follow this and additional works at: [https://digitalcommons.lasalle.edu/ecf\\_capstones](https://digitalcommons.lasalle.edu/ecf_capstones)



Part of the [Databases and Information Systems Commons](#)

---

## Recommended Citation

Byrnes, John, "Development of a Personal Logbook Android Application" (2019). *Economic Crime Forensics Capstones*. 35.  
[https://digitalcommons.lasalle.edu/ecf\\_capstones/35](https://digitalcommons.lasalle.edu/ecf_capstones/35)

This Thesis is brought to you for free and open access by the Economic Crime Forensics Program at La Salle University Digital Commons. It has been accepted for inclusion in Economic Crime Forensics Capstones by an authorized administrator of La Salle University Digital Commons. For more information, please contact [careyc@lasalle.edu](mailto:careyc@lasalle.edu).

# Development of a Personal Logbook Android Application

John Byrnes

La Salle University

Computer Information Science Capstone

## Background

A lifelong hobby of mine is petkeeping. I have several pets at my home and sometimes have difficulty keeping track of important dates and maintenance routines. Veterinarian receipts are placed into a folder after a checkup to update vaccines of the cats and dog and then never looked at again; the vet will send reminders when animals are due for shots, but I would prefer the data be easily accessible and updatable. When I perform aquarium maintenance, like changing filters or checking water parameters, I have a piece of paper that I write the data on, if I remember, so a more modern system would be beneficial. A personal application I can use to log this data will help me stay organized and benefit the health of the animals.

This project is the development of an Android application that will serve as a log that will address the problem described above. I have several years of experience developing software applications as a career but have very limited experience with mobile development. The only situation I have encountered Java and Android development was while taking CIS 629 in the fall of 2016, so this project was intended to be challenging, but rewarding. The purpose is to show how a person with little experience developing Android applications can create something useful and specialized to fit personal needs.

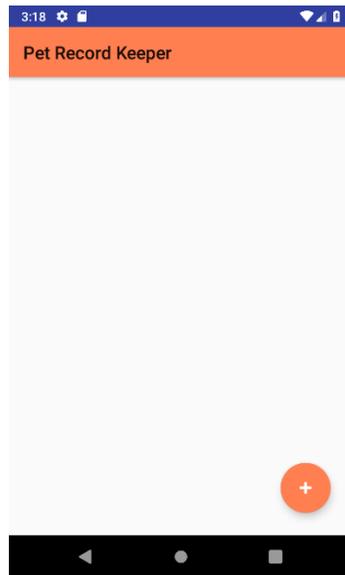
## Downloading and Running the Application

The first step is to visit <https://github.com/jbyrnes10/PetRecordKeeper> and download the code from my repository. GitHub provides a means of version control and is a hosting service so that I can track changes I've made to the code and allow users to use my application easily without publishing to the Google Play Store. Version control is beneficial when working on a team; it is intended to prevent individuals overwriting each other and providing a history so

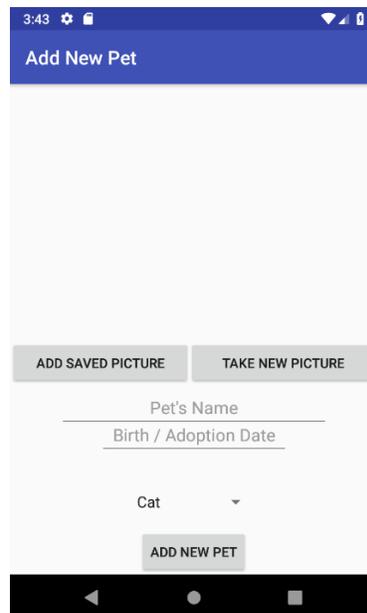
changes can be viewed by all users. This project was done individually, but I still checked in changes periodically as a historical reference; in the future, if I run into issues that I know I had solved before, this will allow me to go back and find my solutions.

The next step is to download Android Studio, which is the official integrated development environment for developing Android applications. The IDE can be downloaded via this link: <https://developer.android.com/studio/>. Installing the IDE can be a lengthy process initially and it seems to notify you often that there are updates available, so just keep reminding yourself that it is a free application. After installation, open the project that was just downloaded from the GitHub repository. Hitting the play button will open a prompt to select a device to run the application on. I debugged my code using the emulator before ever trying it out on my physical device. If you want to get the app on your phone, the first step is to enable debugging. It could be slightly different depending on the device and Android version, but on my phone (Google Pixel), I clicked on “Settings”, then “System”, then “About phone” and scrolled down to the bottom where it says, “Build number”. Tapping “Build number” seven times enables developer settings (there is a visible countdown in case you lose count); this feature being hidden is intentional. If you plug an Android phone into the computer, the physical device should show up as a selectable device when hitting the play button; if it does not, follow these instructions to install the Google USB driver: <https://developer.android.com/studio/run/win-usb>. The application should now be installed on the physical device; if there are any issues, refer to the extensive Android Studio user guide.

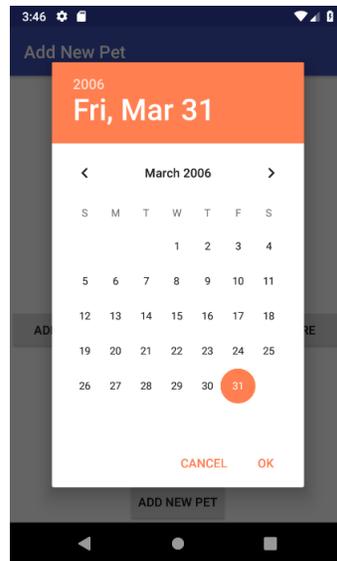
## Using the Application



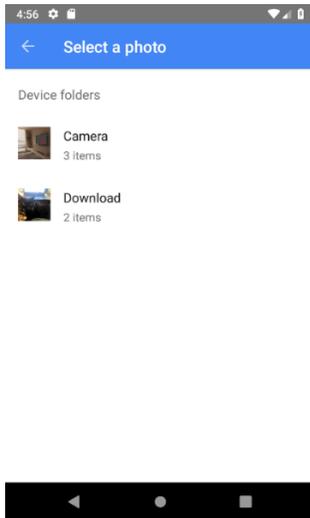
This is the home screen currently. My list of future enhancements includes adding something to eliminate the white space here, but the initial load should be the only time you see a white screen. Click on the plus sign in the bottom right corner to bring up the “Add New Pet” feature.



Click on the text that says, “Pet’s Name” to enable the keyboard to type in a name. When the “Birth / Adoption Date” tapped, the calendar date picker is enabled; clicking “OK” will take the selected date and insert it into the date field.

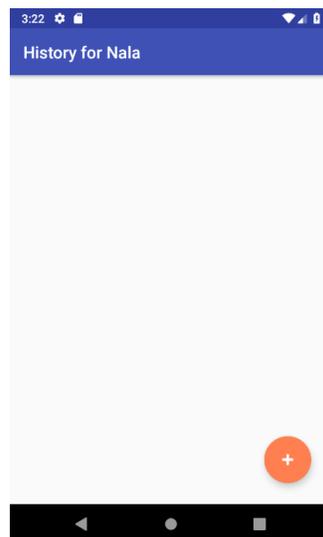


Selecting the “Add Saved Picture” button will show a screen with previously taken photos and downloaded images that can be selected and automatically added to the pet profile. Selecting the “Take New Picture” button will enable the camera on the phone. The image shown below is the default scene the emulator uses; holding the “Alt” button and moving the mouse around lets you scan the room and capture a photo (click the camera button, which turns into a check mark upon success). The check mark can then be clicked, and the captured image will be added to the pet profile.

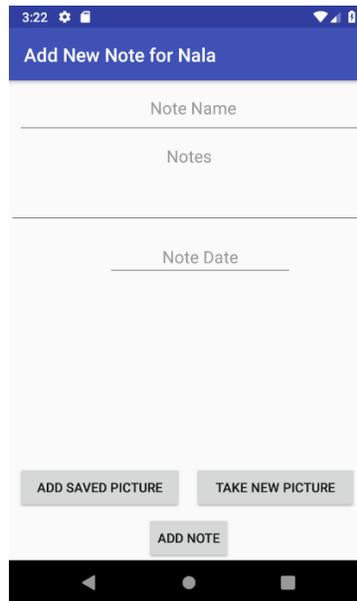


I added saved pictures from my computer to the emulator, so I could test the application using high definition photographs instead of the default images. After spending a lot of time researching how to do this, I discovered how easy it is to add files; finding a file on the computer and dragging and dropping it into the emulator will copy it into the virtual SD card. The files will not show up until you close the emulator and restart it. I chose to add a downloaded image from the emulator to the pet profile.

Tapping the “Add New Pet” button will save the profile to the database and redirect the application to the pet history screen, with a blank slate.

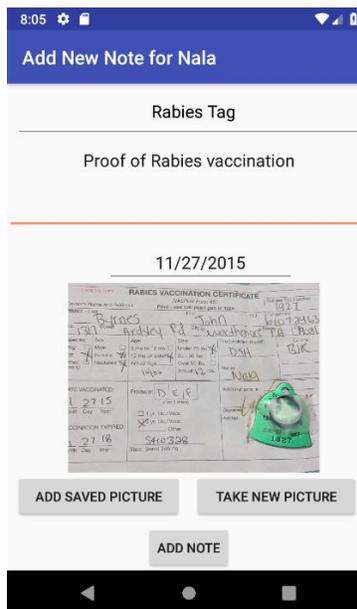


The plus sign button on this screen will bring up a form to add a note to the pet's profile.



A screenshot of a mobile application form titled "Add New Note for Nala". The form has a blue header with the title. Below the header, there are three input fields: "Note Name", "Notes" (a multiline text area), and "Note Date". At the bottom of the form, there are three buttons: "ADD SAVED PICTURE", "TAKE NEW PICTURE", and "ADD NOTE". The phone's status bar at the top shows the time as 3:22 and various icons.

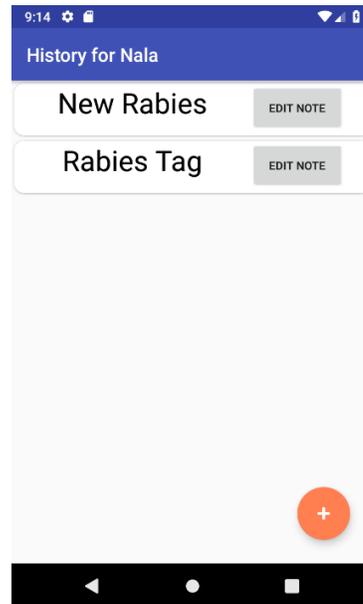
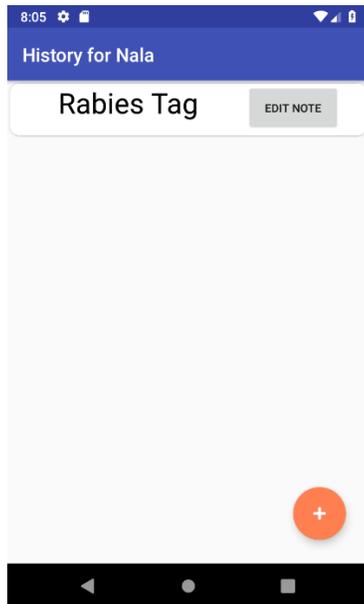
On this screen, you can give the note a name, add multiline text notes, add a date (can be the current date or the date of a veterinarian visit you are documenting), and add a picture or saved image. Below I added a note and picture of a Rabies vaccination tag.



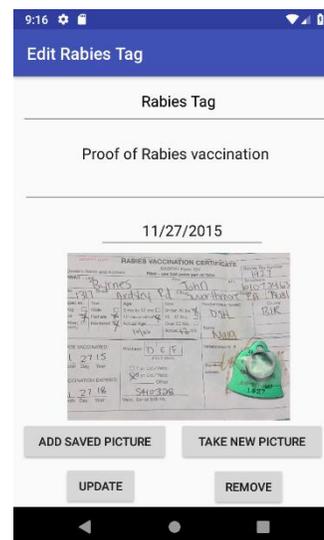
A screenshot of the same mobile application form, now filled with data. The title is "Add New Note for Nala". The "Note Name" field contains "Rabies Tag". The "Notes" field contains "Proof of Rabies vaccination". The "Note Date" field contains "11/27/2015". Below the date field, there is a photograph of a rabies vaccination certificate and a green rabies tag. The certificate has handwritten text including "Rabies Vaccination Certificate", "Brynes", "Jonh", "washington TA", "Bak", "DST", "Nala", "11/27/15", "SHEPHERD", "15.6.18", "2115", "2118", "SHEPHERD". The tag has "11/27" written on it. At the bottom, the "ADD NOTE" button is highlighted. The phone's status bar at the top shows the time as 8:05.

After clicking the "Add Note" button, the data is inserted into the pet history table and the app is redirected to the history list screen, with the newly added note card displayed (displayed on the

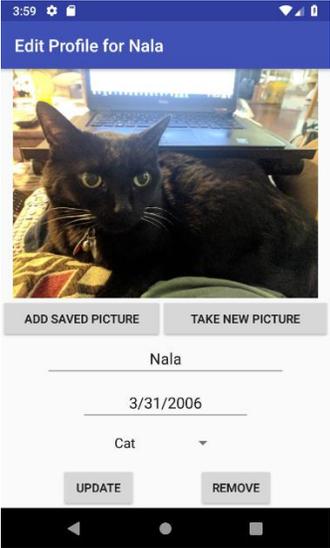
below left screenshot; below right image depicts how the history screen looks after adding a second history note.



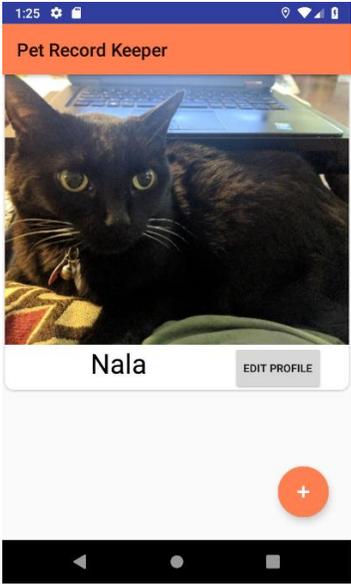
Below to the left is the screen displayed when clicking on the note. To right is the display when the “Edit” button is tapped; each card will display a screen identical to the “Add New Note” screen, with buttons that allow the user to update or delete the note. A copy of the Rabies tag and documentation was chosen for this note. Any images can be added with a note (vet receipts, a picture of a pet with the “cone of shame” around their neck after a surgery, etc.).



Updating or removing a note will redirect to the history list, with the updated changes. If you go back to the home screen, the newly added pet will be visible. Clicking on the image of the pet will show the list of notes and clicking “Edit Profile” will bring up a screen similar to the “Add New Pet” functionality, with the add button being replaced with update and delete functionality.



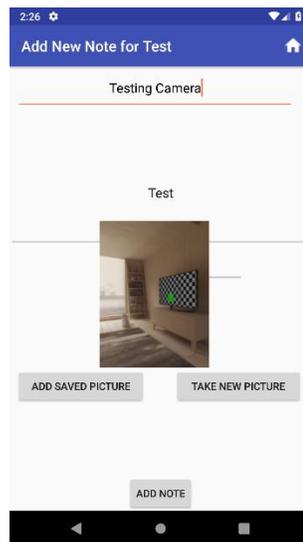
Below to the left is what the home page looks like after adding a pet; to the right is what the home screen looks like when another pet is added.



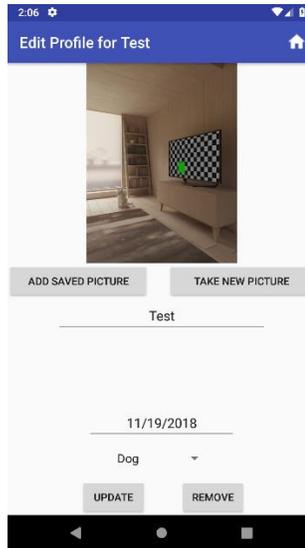
## Current Outstanding Defects

Below is the list of existing code bugs that I am aware of and currently working on. I am testing mostly on an older Nexus 4 phone in the emulator to ensure a wider range of compatibility, but I am also testing on the Google Pixel, which is my current phone.

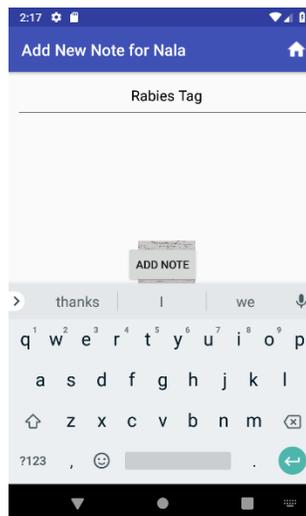
- When adding a saved picture to a history note, the buttons move up and the image overlays the date text. This also happens after clicking the “Edit” button on a history note. This occurs on the Pixel but works well on the Nexus 4. The screenshot is from the Pixel emulator.



- After clicking on the “Edit Profile” button, there is a large amount of whitespace between the pet’s name and the date of birth boxes. This is not the case when adding a new pet. This occurs on the Google Pixel phone (emulator screenshot shown below) but looks fine when using the Nexus emulator.



- When adding a new note, the “Notes” section disappears when the keyboard is active, but the textbox for the name of the note remains, so you can not see what is being typed until you close the keyboard.



## Defects Encountered During Development with Solutions

An important part of this process, as with any software development, is encountering bugs, finding solutions to them, and learning from them for the benefit of future development.

Below is a list of defects I found while testing and how I fixed them; I documented them, so I would not forget about them and then added a solution when I was confident it was fixed.

- PetCardData.java line 57: “cursorindexoutofboundsexception: index -1 requested, with a size of 7”.
  - Solution: A Cursor object will never be null when returning a database object, just have a row count of zero if there is no data matching the query. Adding “cursor.moveToFirst()” to the PetCardData class will go to the first result row of the query.
- When clicking on a card on the home page to pull up the edit card, the app crashes.
  - Solution: I added a try-catch block statement to the EditPetActivity class to see the exception properly. The error message was then easy to see:  
“android.content.ActivityNotFoundException: Unable to find explicit activity class {com.jbyrnes.petrecordkeeper/com.jbyrnes.petrecordkeeper.EditPetActivity}; have you declared this activity in your AndroidManifest.xml?” The error message is self-explanatory: the new activity was not added to the manifest file, so it could not be found. Exceptions are not usually accompanied with an explicit solution like this example.
- When clicking on a card to edit it, it's not pulling back any results from the database.
  - Solution: It was pulling the position of the card and not the id of the table, so it was not finding any rows in the table matching the name and id.
  - This fix was implemented but led to a new error:  
“cursorindexoutofboundsexception: Index -1 requested, with a size of 1.” The solution was the same as a previous bug: adding “cursor.moveToFirst()” solved the out of bounds exception.

- When setting the stored value of the species spinner in the EditPetActivity class, “java.lang.NumberFormatException: For input string: ‘Cat’”.
  - Java does not have an `Array.indexOf()` method, and when setting the spinner selection, it needs an int, not the string text. I had to write a for loop to set the index manually:
 

```
String[] speciesValues = getResources().getStringArray(R.array.species_array);
String savedSpeciesValue = editCard.getSpecies();
int speciesArrayIndex = 0;
for (int i=0; i<speciesValues.length; i++) {
    if (speciesValues[i].equals(savedSpeciesValue)) {
        speciesArrayIndex = i;
        break;
    }
}
```
  - Additional bug after adding this for loop: the species spinner was setting a value correctly, but I didn't set the possible string values from the strings.xml files. The solution was to copy and paste the code from the AddPetActivity that performed this same task.
- Deleting a card from the database on the edit activity does not delete the row from the table.
  - Solution: “`db.delete(PetDatabase.TABLE_PET_PROFILES, whereClause, new String[] {nameExtra, tableId})`” was not returning errors, but was also not deleting the row. The id is a long and that method is expecting all strings. The row properly deletes when I replace it with a straight SQL query, which is the method I prefer anyway.
- When clicking on a pet card on the home screen, I received a `SQLiteException` with the message “no such table PET\_HISTORY”.
  - Solution: I had to change the database version (there is an int declared in the `PetDatabase` class) so that the `onUpgrade` method will run and delete tables and

re-add them. I had changed the table syntax and the database will only run the changes if the database version is different.

- When I added images from my personal phone to the emulator during testing, the images were too large to be stored the way I had been storing the emulator photos.
  - Solution: I had to scale down the images if they are too large. This link provided me with the answer: <https://stackoverflow.com/questions/16954109/reduce-the-size-of-a-bitmap-to-a-specified-size-in-android>.
- “error:failed linking file resources” after I added a new xml layout resource file.
  - Solution: The error message is not helpful; I was getting the error when trying to debug using the emulator. It turns out I put ‘textSize="20"’ instead of “20sp”, which prevented the app from building successfully.

## Future Enhancements

Although the application is in working order, there are several enhancements I would like to implement in the future. Below is the list of these anticipated updates.

- Create an icon for the application so that it is not a default Android icon; currently, in the list of apps on the phone, the name of this application is displayed under the default icon, which launches the app when clicked on.
- Create a home button on the toolbar; there currently is not a quick way to get back to the home screen without hitting back several times or reopening the app. (Completed 11/18)
- Date text boxes only allow the user to select the date from a date picker. This textbox should allow for open text. In order to find my cat’s birthdate, I had to scroll through 12 years to get the correct date.

- Create reminders that automatically save to Google calendar.
- Add error messages when fields are missing when adding a pet or history.
- Add a default message for the initial load of the home page so that it is not a blank screen.
- Add a list of breeds to the pet profile creation.
- Add more pet species to the dropdown list.
- On the initial load of the screen where a pet profile is created, collapse the image view so there is not a large amount of white space; when an image is added, then display the image in the same location it is currently shown.
- Add an alert when deleting a note or a pet profile, in case the user clicked by accident.
- Add styling to the buttons throughout the application; currently they are the default buttons.

## Conclusion

Getting started turned out to be a more difficult task than anticipated. Setting up the IDE and building the basic building blocks for the application were tedious but necessary. Some bugs consumed more time than others, so setting aside enough time for this project and not giving in to frustration was key. Before starting, I recommend creating or thinking about a rough outline of how the app should look and behave. This will change, as it did several times for myself, but having an initial direction is important while getting started. When it comes to testing, choosing an old device and a newer one was a smart choice. Testing all Android devices is not realistic because there are so many, but since this was a personal project, making sure the app works on my phone as well as an older device seemed like the way to go.

Developing an Android application can be overwhelming at times, and it is certainly challenging. A more experienced Java developer may have finished the application earlier or approached it differently, but this project shows that a person with limited experience can still produce a working Android application, customized to fit personal needs.

## References

Android Studio: User guide. Retrieved from

<https://developer.android.com/studio/intro/?nav=true>.

Annuzzi Jr., J., Darcey, L., & Conder, S. (2016). *Introduction to Android Application Development, Fifth Edition*. New York: Addison-Wesley.